EECS3342 System Specification and Refinement
Fall 2026

Digest of Lecture 7, October 01

Relational Overriding, Functions, and Application

Jackie Wang

How to use this Digest

Use this digest as a **roadmap and checklist**. It highlights the main ideas, important examples, common pitfalls, practice targets, and where to return in the course materials. It does **not** replace attending or reviewing the session, using the recording when needed, studying Jackie's annotated iPad notes and other official course materials, completing assigned work, or practising independently. Use it to identify what you understand, what you need to revisit, and what you should be able to do on your own.

If time is limited, start with the sections most relevant to your current review and return to the rest later; selective use is better than skipping the digest because it looks long.

This is an AI-assisted study roadmap reviewed by Jackie and grounded in the session transcript and Jackie's annotated iPad notes. Like any course document, it may still contain an error or omission. If something seems technically doubtful or inconsistent with the original course materials, return to those materials and ask Jackie for clarification rather than treating the digest as final authority.

Table of Contents

Main Ideas 2

Roadmap 2

Key Terms, Notation, and Structures 5

Key Examples and Demonstrations 5

Common Pitfalls and Misconceptions 6

What You Should Be Able to Do 7

Suggested Practice 8

Where to Review 8

Closing Reminder 9

Main Ideas

- **Relational overriding can be understood as targeted replacement.** When r is overridden by a relation t, the pairs of r whose first components lie in $\mathrm{dom}(t)$ are replaced by the pairs in t. This agrees with the formal domain-subtraction-plus-union definition from the previous lecture.
- **Relational image can be reconstructed from restriction and range.** First restrict attention to pairs whose first components are in the input set; then take the range of that restricted relation. The intermediate restriction is still a relation, so applying ran is necessary to obtain the image set.
- **A function is a relation satisfying a uniqueness property.** Each domain value can map to at most one range value. To disprove functionality, it is enough to find two pairs with the same first component and distinct second components.
- **Partial versus total functions is fundamentally a domain question.** A partial function may leave some source values undefined; a total function must have domain exactly equal to the source set. Jackie's nested-set diagram places total functions inside partial functions, which are themselves inside relations.
- **Relational image and functional application behave differently.** Image notation returns a set and can return the empty set when no mapping is present. Functional application returns a single value and is undefined when the argument is outside the function's domain; in Rodin this produces a well-definedness proof obligation.

Roadmap

1. Programming-test priorities

Jackie reminded the class that the only programming test is on **Wednesday, October 7**, during each student's **enrolled lab session**. Students should check their lab time and attend only their own session. The PDF guide and practice tests are available, including the Fall 2024 test, and the Event-B Summary document will be available during the test. Jackie recommended preparing thoroughly before using the Fall 2024 test as a realistic readiness check. Lab 1 and Lab 2 solutions were also available for review.

2. A second interpretation of relational overriding

The session resumed the overriding example from Lecture 6. For

$$t = \{(a, 3), (c, 4)\},$$

Jackie first identified $\mathrm{dom}(t) = \{a, c\}$. In the original relation r, every pair whose first component is a or c is the part that may change. The alternative interpretation is therefore:

keep the rest of r unchanged, remove all pairs of r whose first components are in $\mathrm{dom}(t)$, and replace that part by exactly the pairs in t.

This is the same result as the formal rule from Lecture 6: take t and union it with the part of r left after domain subtraction by $\mathrm{dom}(t)$.

Jackie then connected overriding to the Lab 1 transfer event. Because an Event-B variable is assigned at most once in one event, two balance updates should be consolidated into one action such as

$$b := b <+ \{\mathrm{ACC1} \mapsto v_1, \mathrm{ACC2} \mapsto v_2\}.$$

The right-hand relation contains the replacement pairs for the accounts whose balances change.

3. Expressing relational image using other operators

Jackie asked which of the four restriction/subtraction operators moves in the right direction for computing $r[s]$. Domain restriction is the relevant starting point because image computation first isolates those pairs whose *first* component lies in s.

A type check is crucial. Domain restriction returns a *relation*, whereas $r[s]$ is a *set of target values*; therefore the two cannot simply be set equal. The corrected property developed on p. 4 of Jackie's annotated iPad notes is

$$r[s] = \mathrm{ran}(s \lhd r),$$

where $s \lhd r$ denotes the domain-restricted relation. The range operator is what converts the restricted relation into the target set represented by the image.

4. Functional property: proof, disproof, and contrapositive

The functional property from Lecture 6 was revisited. Jackie's annotated iPad notes state

$$\mathrm{isFunctional}(r) \Leftrightarrow \forall s, t_1, t_2 \bullet (s \in S \land t_1 \in T \land t_2 \in T) \Rightarrow \big((s, t_1) \in r \land (s, t_2) \in r \Rightarrow t_1 = t_2\big).$$

Its intuitive meaning is that each domain value maps to **at most one** range value.

The smallest function discussed in class is the empty relation: there is no witness that can violate the universal functional condition. For a nonempty relation, Jackie described proving functionality by checking that every domain value occurring in the relation maps to at most one range value. To disprove functionality, find two pairs

$$(s, t_1), (s, t_2) \in r \qquad \text{with} \qquad t_1 \neq t_2.$$

The example $\{(a, 1), (b, 2), (a, 3)\}$ is therefore not a function.

Jackie also rewrote the inner implication contrapositively:

$$t_1 \neq t_2 \Rightarrow \neg\big((s, t_1) \in r \land (s, t_2) \in r\big).$$

This gives a practical repair idea: conflicting pairs cannot both remain in a functional relation. Removing at least one of $(a, 1)$ and $(a, 3)$ removes that particular violation.

5. Partial functions versus total functions

The second half of the lecture developed the distinction from the definitions. For a candidate relation r from source S to target T, the notes use the following conditions:

$$\begin{aligned} &\text{partial function:} && \text{isFunctional}(r) \land \text{dom}(r) \subseteq S, \\ &\text{total function:} && \text{isFunctional}(r) \land \text{dom}(r) = S. \end{aligned}$$

The distinction is therefore whether the function may leave some values in S undefined. If $\text{dom}(r) \subset S$, at least one source value has no mapping. If $\text{dom}(r) = S$, every source value has a range value and the function is total.

Jackie's nested-set visualization on p. 7 organizes the concepts as *relations* containing *partial functions*, with *total functions* inside the partial-function set. The lecture emphasized: every total function is also a partial function, but not every partial function is total.

6. Membership versus subset for sets of functions

The function-arrow notation denotes a *set of functions*, not one function. This makes the choice between $\in$ and $\subseteq$ a type question. If the left-hand side is one candidate function, membership is appropriate. If the left-hand side is itself a set containing several candidate functions, subset notation is appropriate.

Jackie stressed that using membership where the left side is a set of functions is not merely a false mathematical claim in Rodin; it is a type error and does not compile. The examples on pp. 7–8 of Jackie's annotated iPad notes make the set-versus-element distinction explicit.

7. Relational image versus functional application

Finally, Jackie compared two ways to use the same partial function

$$f = \{(3, a), (1, b)\}.$$

Treating f as a relation gives

$$f[\{3\}] = \{a\}, \qquad f[\{1\}] = \{b\}, \qquad f[\{2\}] = \varnothing.$$

Treating f as a function gives

$$f(3) = a, \qquad f(1) = b,$$

while $f(2)$ is undefined because $2 \notin \text{dom}(f)$.

The Rodin consequence is important. With a correctly typed input, relational image returns a set and remains defined even when there is no matching pair. Functional application $g(c)$, however, requires

$$c \in \text{dom}(g),$$

and Rodin generates a proof obligation requiring that condition to be established.

Key Terms, Notation, and Structures

Relational overriding
The Event-B operation written in Rodin as `r <+ t`. Its replacement interpretation is: retain the unaffected part of r, replace the part whose first components lie in $\mathrm{dom}(t)$, and use exactly the pairs in t for that domain.

$r[s]$ — relational image
A set of target values reached from source values in s. In this lecture, Jackie connected it to domain restriction and range:

$$r[s] = \mathrm{ran}(s \lhd r).$$

Functional property
The uniqueness condition that each domain value maps to at most one range value. Two pairs with the same first component and distinct second components form a violation witness.

Partial function
A functional relation that may leave some values of the source set undefined. The defining domain condition used in class is $\mathrm{dom}(r) \subseteq S$. Use the exact Event-B partial-function arrow shown on pp. 7–8 of Jackie's annotated iPad notes when entering the notation in Rodin.

Total function
A functional relation whose domain is exactly the source set: $\mathrm{dom}(r) = S$. Every source value therefore has a mapped target value.

Function-set membership versus subset
Use $\in$ when the left-hand side is one candidate function and the right-hand side is a set of functions. Use $\subseteq$ when both sides are sets of functions.

Functional application
For a function f, $f(x)$ denotes a single target value and requires $x \in \mathrm{dom}(f)$. If the argument is outside the domain, the application is undefined.

Proof obligation (PO)
A condition Rodin requires to justify correctness or well-definedness. In the lecture's final example, applying $g(c)$ creates the obligation $c \in \mathrm{dom}(g)$.

Key Examples and Demonstrations

1. Overriding as replacement

Setup. Override the working relation r by $t = \{(a, 3), (c, 4)\}$.

Main point. First compute $\mathrm{dom}(t) = \{a, c\}$. Remove from r all old pairs beginning with a or c, then insert exactly the pairs in t. This is the visual replacement interpretation on p. 3 of Jackie's annotated iPad notes and agrees with the formal domain-subtraction rule.

What to practise. Reproduce the result once from the formal definition and once from the replacement interpretation, and explain why the two methods agree.

2. Lab 1 transfer update

Setup. A transfer changes two entries of the balance function b.

Main point. Event-B does not allow separate assignments to the same variable inside one event. Jackie used overriding to combine the two account updates into one assignment of the form

$$b := b <+ \{\mathrm{ACC1} \mapsto v_1, \mathrm{ACC2} \mapsto v_2\}.$$

What to practise. Return to the released Lab 1 solution and identify which part is the old function, which relation contains the replacement pairs, and why a single overriding assignment is used.

3. Classifying relation examples

Setup. Jackie's notes use source $S = \{1, 2, 3\}$ and target values such as $\{a, b\}$.

Main point. The examples distinguish three cases:

- $\{(2, a), (1, b)\}$: functional but not total because the domain is only $\{1, 2\}$;
- $\{(2, a), (3, a), (1, b)\}$: total because it is functional and its domain is all of S;
- $\{(2, a), (1, b), (3, a), (1, a)\}$: not a function because the source value 1 maps to both a and b.

What to practise. For each candidate, first check the functional property and only then check whether its domain equals the full source set.

4. Image versus application

Setup. Use $f = \{(3, a), (1, b)\}$.

Main point. The relational image of $\{2\}$ is the empty set, while functional application at 2 is undefined. This is not merely a notation difference: the two operations have different result types and different well-definedness requirements.

What to practise. Reproduce all three examples on p. 9 of Jackie's annotated iPad notes and explain why the p. 10 Rodin application example needs a domain-membership proof obligation.

Common Pitfalls and Misconceptions

- **Thinking overriding depends on the written order of pairs.** Relations are sets, so pair-listing order does not change the relation.
- **Assigning to the same Event-B variable more than once in one event.** Jackie used overriding specifically to consolidate several changes to one relation/function variable into a single assignment.

- **Equating an image directly with a restricted relation.** The image is a set of target values, whereas domain restriction still returns a relation. Apply ran to obtain compatible types.
- **Checking totality before checking functionality.** A relation whose domain covers all of S is still not a total function if one source value maps to more than one target value.
- **Reversing the partial/total containment relationship.** The lecture emphasized that every total function is partial, but a partial function may leave some source values undefined and therefore need not be total.
- **Using $\in$ when the left-hand side is itself a set of functions.** In the lecture example, that is a Rodin type error; use $\subseteq$ when comparing a set of candidate functions with a set of all functions.
- **Confusing empty relational image with undefined function application.** For an argument outside the domain, $f[\{x\}]$ can be $\varnothing$, while $f(x)$ is undefined.
- **Using only informal labels such as "general" or "restrictive" on a test.** Jackie used those words only as study intuition; formal answers should state the functional and domain conditions.

What You Should Be Able to Do

After this session, you should be able to:

- explain relational overriding using both the formal domain-subtraction view and Jackie's replacement interpretation;
- use overriding to explain why several updates to one Event-B relation or function variable can be written as one assignment;
- derive relational image from domain restriction followed by ran, and explain the type mismatch in the incomplete restriction-only equality;
- state the functional property and prove or disprove it using the source-grounded strategies from the lecture;
- rewrite the inner functional implication using its contrapositive and use that form to explain how a conflicting relation can be repaired;
- distinguish partial from total functions by their domain conditions and classify the concrete examples from Jackie's annotated iPad notes;
- choose correctly between $\in$ and $\subseteq$ when working with one function versus a set of functions;
- distinguish relational image $f[\{x\}]$ from functional application $f(x)$, including empty-set versus undefined outcomes; and
- explain why Rodin requires a proof obligation $x \in \mathrm{dom}(f)$ for functional application.

Suggested Practice

- Prepare for the October 7 programming test using the PDF guide and the released practice tests. As Jackie recommended, use the Fall 2024 test as a serious readiness check after you have prepared as much as possible, and verify your enrolled lab-session time.
- Revisit the two interpretations of overriding on pp. 3–4 of Jackie's annotated iPad notes. Starting from the same r and t, reproduce the result both ways without looking at the handwritten result first.
- Review the released Lab 1 solution and locate the transfer action that uses overriding. Explain why two separate assignments to b would not be allowed in the same event.
- Rework the proof/disproof material on pp. 5–6: identify a violation witness for the non-functional example, then use the contrapositive to explain why removing one conflicting pair repairs that violation.
- Reclassify the partial/total/not-a-function examples on p. 7 by checking functionality first and domain coverage second. Then revisit the set-versus- element notation on p. 8.
- Recompute the relational-image and functional-application examples on p. 9, and connect the undefined application case to the proof obligation on p. 10.
- Study ahead before the next class. Jackie said the math review would continue briefly with injective, surjective, and bijective functions before the course moves on to the bridge-controller material.

Where to Review

- **Jackie's annotated iPad notes, p. 3:** the alternative replacement interpretation of relational overriding and the Lab 1 transfer connection.
- **Jackie's annotated iPad notes, p. 4:** the algebraic properties for overriding and relational image, including $r[s] = \mathrm{ran}(s \lhd r)$.
- **Jackie's annotated iPad notes, pp. 5–6:** the functional property, the empty relation as the smallest function, proof/disproof strategies, the contrapositive, and the repair of the conflicting-pair example.
- **Jackie's annotated iPad notes, pp. 7–8:** partial versus total functions, the nested-set visualization, the classification examples, and the membership-versus-subset type distinction.
- **Jackie's annotated iPad notes, pp. 9–10:** relational image versus functional application and the Rodin well-definedness/proof-obligation comparison.
- **Released Lab 1 solution:** the transfer-event use of overriding for updating more than one account balance through one assignment.

- **Lecture recording:** return to the overriding replacement explanation, the proof/disproof discussion, and the partial-versus-total function development if the handwritten progression needs Jackie's spoken reasoning.

Closing Reminder

Use this digest as a roadmap for active review, not as a substitute for the learning process itself. Attend Jackie's in-person sessions and pay attention to how he develops ideas and works through the important reasoning step by step. If a transition, explanation, example, or reasoning step remains unclear afterwards, return to the recording to fill that gap.

For structured review, start with Jackie's annotated iPad notes because they most closely reflect the actual flow and development of the class. Work through them in manageable logical units. After each logical unit, study the corresponding part of the presentation slides alongside it. Jackie does not normally teach by reading the slides line by line, so the slides may contain relevant supporting material that was not explicitly read aloud.

Finally, choose a suitable selection of the important reasoning from this session and work through it again on your own. The goal is not to memorize Jackie's recorded steps, but to determine whether you can reproduce the reasoning from your own understanding and explain why the important steps work.